Proceedings of the 1998 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Victoria, B.C., Canada • October 1998

# Landmark identification and tracking in natural environment

Rafael Murrieta-Cid, Maurice Briot and Nicolas Vandapel
{murrieta,briot,vandapel}@laas.fr
L.A.A.S.-C.N.R.S.
7 Av. du Colonel Roche, 31077 Toulouse Cédex 4, France

## Abstract

*This paper deals with the development of the use of visual function in order to add useful information for tasks of a mobile robot roving in natural environments. We have proposed and implemented the nominative region model which indicates every region's nature in an image. A color segmentation algorithm provides a synthetic description of the scene. Regions obtained from the segmentation stage are then characterized by their color and texture and afterwards identified in order to obtain their nature (grass, rocks, ...). Probabilistic methods are used to determine the nature of current elements in the environment. Then, one specific landmark is chosen according to its nature and shape and this representation is tracked through an image sequence.*

## 1 Introduction

Our study takes place in the perception for autonomous mobile robots evolving in outdoor environments. The classical line of research in perception for mobile robots is based on 3-D information, obtained by a laser ranger finder or a stereoscopic system. However depth information is not enough to get a complete description of the environment. Other information such as the nature of the elements in the scene needs to be taken into consideration.

In this paper we present our ongoing research on three specific problems of artificial vision: i) color segmentation ii) regions characterization and identification by using color and texture and iii) visual tracking.

We will only outline the main features of our approach for each problem and show some experimental results. Our approach consists of several phases executed sequentially, the cooperation among them is possible. The results of the one previous phase could be checked by the current phase and if necessary corrected. The main phases are:

- region extraction: firstly, the color image is segmented to obtain the principal regions of the scene.
- region characterization: each region of the scene is characterized by its color and texture.
- regions identification: the nature (class) of the elements (regions) in the scene is obtained by comparing a vector of features with a database composed of different classes, issued from a learning process. The database is a function of the environment type. Here, we have chosen 4 classes which correspond to the main elements in our environment: grass, sky, tree and rock. After that, the regions of the same nature are merged and the consistency of the results is verified through context.
- automated selection and tracking of a landmark: an appropriated landmark is selected automatically by taking into account its nature and shape. A model of the landmark is then tracked through an image sequence.

These phases are performed on images of different resolutions. The color segmentation has to quickly give a synthetic representation of the scene, so this stage is done with images of low resolution (images of size 128x128 pixels). The phase of characterization gives better results using the larger resolution (images 512x512). The fusion of regions is brought out to the same resolution as that of the segmentation. Landmark selection and tracking are done in images of size 256x256.

## 2 Color segmentation

The segmentation of natural outdoor scenes is a very difficult task due to the huge variety of images and their complexity.

The main goal of this phase is to achieve a segmentation of large regions corresponding to the main

0-7803-4465-0/98 $10.00 © 1998 IEEE

elements of the scene. The regions obtained from this process can be used as a good input for an identification task.

In previous works [6, 7] we have tested several color spaces. In our study, the best color segmentation was obtained by using the $I_1, I_2', I_3'$ space defined as [9]: $I_1 = \frac{R+G+B}{3}$, $I_2' = (R - B)$, $I_3' = \frac{2G-R-B}{2}$. This space allows to obtain fewer regions and good quality of segmentation even for complex images.

We have developed a segmentation algorithm [7], which is a combination of two techniques: the characteristic feature thresholding or clustering, and the region growing technique. The method tries to do the grouping in the spatial domain of the image but it also uses the attribute space (color space).

The advantage of this method is that it allows the merging process independently of the beginning point and the scanning order of the adjacent regions.

## 3 Region characterization

Each region of the scene is characterized by its color and texture. The texture operators are based on the sum and difference histograms, this type of texture measure is an alternative to the usual co-occurrence matrices used for texture analysis. The sum and difference histograms used conjointly are nearly as powerful as co-occurrence matrices for texture discrimination. The advantage of this texture analysis method over co-occurrence matrices is the decrease in computation time and memory storage required.

Statistical information can be extracted from these histograms. We have used 6 texture features computed from the sum and difference histograms, these features are [10]: Mean, Variance, Energy, Entropy, Contrast and Homogeneity.

In addition, to the 6 texture features, the statistical means of $I_2', I_3'$ are used to characterize each region in the image. In order to reduce the dependency on intensity changes in the identification step, the intensity component was not used.

## 4 Region identification

The nature (class) of the elements (regions) in the scene is obtained by comparing a vector of features with a database composed of different classes, which was obtained from a learning phase. Two classification techniques are used and compared. The Bayesian classification and a hierarchical classifier based on the concept of average mutual information.

### 4.1 The Bayesian classification

The first tested probabilistic approach is the Bayesian rule, defined as:

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{\sum_{i=1}^{n} P(X \mid C_i)P(C_i)}$$

Where: $P(C_i)$ is the *a priori* probability that a region belongs to the class $(C_i)$. $P(X \mid C_i)$ is the class conditional probability that the region is $X$, given that it belongs to class $(C_i)$. $P(C_i \mid X)$ is the *a posteriori* conditional probability that the region's class membership is $C_i$, given that the region is $X$.

We have assumed equal *a priori* probability. In this case the computation of the *a posteriori* probability $P(C_i \mid X)$ can be simplified and its value depends solely on $P(X \mid C_i)$.

The value of $P(X \mid C_i)$ is estimated by using the k-nearest neighbor method. A sample $X$ will be assigned to the class $C_i$ whose $k - th$ nearest neighbor to $X$ is closest to $X$ than to any other training class.

The Bayesian classification does not need the partitioning of the feature space and integrates the different factors into a formal and rigorous frame. However, this method requires the computation of the *a posteriori* conditional probability for each class.

### 4.2 Hierarchical classifier

The second tested classification technique is based on an algorithm for the partitioning of the feature space [8]. This algorithm has inherent feature selection capability.

The algorithm gives rise to a locally optimum decision tree by maximizing the amount of average mutual information obtained at each partitioning step.

The average mutual information obtained about a set of classes $C_k$ from the observation of an event $X_k$, at a node $k$ in a tree $T$ is defined as:

$$I_k(C_k, X_k) = \sum_{C_k} \sum_{X_k} p(C_{ki}, X_{kj}) \cdot \log_2 [\frac{p(C_{ki} \mid X_{kj})}{p(C_{ki})}]$$

Event $X_k$ represents the measurement value of a feature selected at node $k$ and has two possibles outcomes; measurement values greater or smaller than a threshold associated with that feature at that node. In order to do the partitioning of the feature space, we test the Shannon's entropy $H = p(C_{ki} \mid X_{kj}) \cdot \log p(C_{ki} \mid X_{kj})$ for the different classes at a node $k$ in a tree $T$. If this entropy is greater than a given threshold the node is further split, otherwise the division is stopped for this node.

In addition to the partitioning of the feature space, we are defining security areas for identification. These areas are determined by using the statistical mean and the standard deviation of the features selected at each terminal node. Areas of the feature space that fall outside of the confidence borders could be interpreted as regions of non-classification.

The decision trees are attractive for the following reasons: global complex decision areas can be approximated by the union of simpler local decision areas at various levels of the tree. In a tree classifier a sample is tested against only certain subsets of classes, in addition it has the flexibility of choosing different subsets of features at different internal nodes of the tree. Consequently, it allows the elimination of unnecessary computations. Nonetheless, this method also has some drawbacks. Two internal nodes that contain at least one common class can cause the number of terminals to be much larger than the number of actual classes and thus increase the search time and memory space requirements.

## 5 Fusion of regions and coherence of the model

In this phase of the process, each region in the image has a class associated (nature). These regions were obtained from the color segmentation phase. However, the segmentation results in large regions, the regions do not always correspond to real objects in the scene. Sometimes a real element is over-segmented, consequently a fusion phase becomes necessary. In this step, connecting regions belonging to the same class are merged.

The coherence of the model is tested by using the topological characteristics of the environment [6]. Possible errors in the identification process could be detected and corrected by using contextual information (i.e. grass cannot be surrounded by sky regions).

## 6 The target tracking method

The target tracking problem has received a great deal of attention in the computer vision community over the last years. Several techniques have been reported in the literature, and a variety of features have been proposed to perform the tracking [5, 2].

We are using a method able to compute the motion of an object in the image due to the motion of the sensor or the motion of the object.

The target's motion in the 2D image can be decomposed into two parts:

- A two-dimensional motion in the image, corresponding to the change of the target's position in the image space.
- A two-dimensional shape change, corresponding to a new aspect of the target.

The tracking is done using a comparison between an image and a model. The model and the image are binary elements extracted from a sequence of gray level images using an edge detector similar to [1].

To measure the resemblance of an image with the model we use the partial Hausdorff distance.

Given two sets of points $P$ and $Q$, the Hausdorff distance is defined as:

$$H(P,Q) = \max(h(P,Q), h(Q,P))$$

where

$$h(P,Q) = \max_{p \in P} \min_{q \in Q} \| p - q \|$$

and $\| \cdot \|$ is some norm for measuring the distance between two points $p$ and $q$. The Hausdorff distance is the maximum among $h(P,Q)$ and $h(Q,P)$.

By computing the Hausdorff distance in this way we obtain the most mismatched point between the two shapes compared; consequently, it is very sensitive to the presence of any outlying points. For that reason it is often appropriate to use a more general rank order measure, which replaces the maximization operation with a rank operation. This measure (partial distance) is defined as [3]. $h_k = K^{th}_{p \in P} \min_{q \in Q} \| p - q \|$. Where $K^{th}_{p \in P} f(p)$ denotes the $K^{-th}$ ranked value of $f(p)$ over the set $P$.

The term $h_{k1}(P,Q)$ is the unidirectional partial distance from the model to the image, and $h_{k2}(Q,P)$ is the unidirectional partial distance from the image to the model. Where $P = M_t$ is the model and $Q = I_t$ is the image or region of the image given at $t$ time of one sequence. The maximum of these two values defines the partial Hausdorff distance.

### 6.1 Finding the model position

The first task to accomplish is to define the position of the model $M_t$ in the next image $I_{t+1}$ of the sequence. The search for the model in the image (or image's region) is done in some direction selected. We are using the unidirectional partial distance from the model to the image to achieve this first step.

It is possible to identify the set of translations of $M_t$ such that $h_{k1}(M_t, I_{t+1})$ is no larger than some value

$\tau$, in this case there may be multiple translations that have essentially the same quality [4]. However, rather than computing the single translation giving the minimum distance or the set of translations, such that its correspond $h_{k1}$ is no larger than $\tau$, it is possible to find the first translation, such that its associated $h_{k1}$ is no larger than $\tau$, for a given search direction., in this way the computing time is significantly reduced.

### 6.2 Checking target position and building the new model

Having found the position of the model $M_t$ in the next image $I_{t+1}$ of the sequence, we now have to build the new model $M_{t+1}$ by determining which pixels of the image $I_{t+1}$ are part of this new model.

The model is updated by using the unidirectional partial distance from the image to the model. The new model is defined as:

$$M_{t+1} = \{q \in I_{t+1} \mid h_{k2}(I_{t+1}, g(M_t)) < \delta\}$$

Where $g(M_t)$ is the model at the time $t$ under the action of the translation $g$, and $\delta$ controls the degree to which the method is able to track objects that change shape.

The tracking of the model is successful if:

$$k1 > fM \mid h_{k1}(M_t, I_{t+1}) < \tau$$

and

$$k2 > fI \mid h_{k2}(I_{t+1}, g(M_t)) < \delta \ ,$$

in which $fM$ is a fraction of the number total of points of the model $M_t$ and $fI$ is a fraction of image's point of $I_{t+1}$ superimposed on $g(M_t)$.

### 6.3 Our contributions over the tracking method

The target tracking method presented in this paper is based on the one introduced in [3] and [4]. This section enumerates some extensions that we have made over the general method.

Firstly, we are using an automatic identification method in order to select the initial model. This method uses several attributes of the image such as color, texture and shape. Secondly, only a small region of the image is examined to obtain the new target position, as opposed to the entire image. In this manner, the computation time is decreased significantly. The idea behind a local exploration of the image is that if the execution of the code is quick enough, the new target position will then lie within a vicinity of the previous one. In this way, the robustness of the method is increased to handle target deformations, since it is less likely that the shape of the model will change significantly in a small $\delta t$. Finally, instead of computing the set of translations of $M_t$, such that $h_{k1}(M_t, I_{t+1})$ is no larger than some value $\tau$, we are finding the first translation whose $h_{k1}(M_t, I_{t+1})$ is less than $\tau$. This strategy significantly decreases the computational time.

## 7 Cooperation between the nominative model of the scene and the visual tracking

We underline that the nominative model of the scene is used to select automatically an appropriated landmark. This approach allows the selection of a landmark based on its nature and shape.

When several elements having the same nature are present in the scene, the nominative model of regions could be used to select one according to its two-dimensional representation. i.e., the longest region belonging to the class rock, present in the image. It is also possible to track portions of landmark to decrease the computation running time of the tracking process. One criteria is being here to select the element with the largest elongation when there are several elements of the same nature. This criteria is as follows: The first step is to select the longest region in the image. The major vertical axis of the object is found, and a window is constructed around it. The window width is determined as a fraction of the size of the major vertical axis, only the points belonging to the region of the class chosen and falling within the window are taken into consideration. In addition very narrow elements are avoided.

## 8 Experimental results

To show the construction of the nominative model, we present this process in a image. Figure 1 shows the original image. Figure 2 shows the color image segmentation and the identification of the regions. Labels in the images indicate the nature of the regions: (R) rock, (G) grass, (T) tree and (S) sky. The identification in this case was performed by using the hierarchical classifier.

The Region at the top right corner of the image was identified as grass. However, this region is placed out of the confidence borders defined for this class, in this case the system can correct the mistake by using

Figure 1: Original image

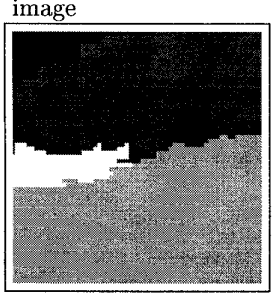

Figure 3: Final model

Fig 5

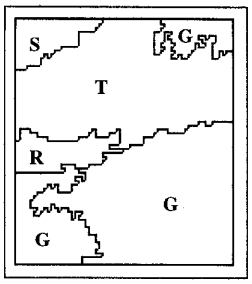


Figure 2: Segmentation and Identification

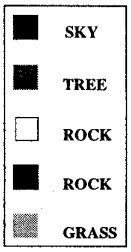


Figure 4: classes

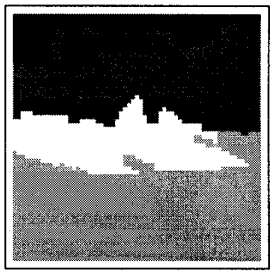

Fig 6

Fig 7

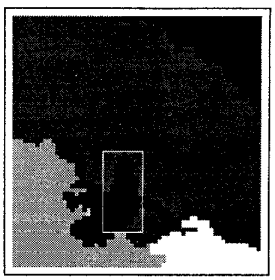

Fig 8

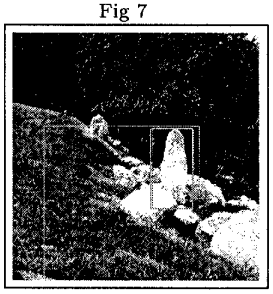

Fig 9

Fig 10

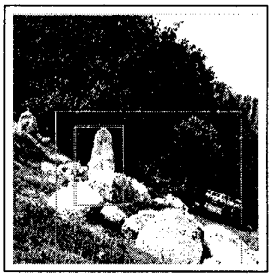

Fig 11

Fig 12

contextual information; this region is then relabeled as tree, figure 3 shows the final model of this scene. Figure 4 shows the gray levels used to label the classes, two gray levels were used to label the class rock in order to show the capability to select one as a target according to its two-dimensional representation. Figures from 5 to 8 show the originals images and their nominative model.

The general results of the identification were as follows: The database was generated from 40 images. The identification was performed over 20 images, none of which were included in the training set.

Table 1 shows the experimental results obtained by using the Bayesian classification, Table 2 shows the results by using the hierarchical classifier. The results of identification in both tables do not include the correction by contextual information; whether this correction is done the errors are almost totally eliminated.

Bayesian classification gives somewhat better results than the hierarchical classifier, however the hierarchical classifier eliminates computations by allowing the selection of different subsets of features at different internal nodes of the tree.

The tracking method was implemented in C, the computation running time is dependent on the region size examined to obtain the new target position. For video sequences the code is capable of processing a frame in about 0.3 seconds for a video image of (256 X 256 pixels). The construction of the nominative model of the scene is done in about 3.5 seconds. The computer employed in these experimentations was a SPARC 20.

Figure 7 shows the original image, figure 8 shows the automatic selection of a landmark based on its nature and shape. In this case a portion of the rock

| Nature | Number of regions | Number of regions identified | % of success |
|---|---|---|---|
| Tree | 46 | 43 | 93% |
| Ski | 26 | 25 | 96% |
| Grass | 29 | 26 | 89% |
| Rock | 34 | 34 | 100% |
| Total | 135 | 128 | 94% |

Table 1: results for the Bayesian classification

| Nature | Number of regions | Number of regions identified | % of success |
|---|---|---|---|
| Tree | 46 | 40 | 91% |
| Ski | 26 | 25 | 96% |
| Grass | 29 | 25 | 86% |
| Rock | 34 | 34 | 100% |
| Total | 135 | 124 | 91% |

Table 2: results for the hierarchical classifier

having the largest elongation is selected as the target. The selection criteria described previously is utilized here. Figures 9, 10, 11 and 12 show the tracking of a rock, this rock is marked in the figure with a boundary box. Another larger boundary box is used to delineate the region of examination.

## 9 Conclusion and future work

A mobile robot must have complete perceptual capabilities to be able to performed a complex task. Computer visual techniques can provide useful knowledge about the environment. In order to obtain this knowledge different image processing are necessary from pixel correlation up to high-level operations such as image understanding. Even though our approach consists of several phases executed in sequence, the cooperation among them is possible. Over-segmentation and identification errors can be corrected by using contextual information from the environment. Regions with several elements of a different nature can be detected by using an homogeneity measure of *a posteriori* probability of the region's class membership, such as Shannon's entropy. These regions can be eventually re-segmented.

In terms of adding functionality to the system, there are some possible extensions: first, we plan to use the nominative model of regions in order to detect a tracking drift. Since the speed of computation of the nominative model is slower than the tracking process, the nominative model of regions will be computed to a smaller frequency. The goal is to perform a double-check by using the nature of the target in addition to the partial Hausdorff distance to detect possibles tracking errors. Second, we would also like to consider the case of multiple targets.

## References


[1] J. Canny, A computational approach to edge detection, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6), 198

[2] Yue Du, A color projection for fast generic target tracking, *Int. Conf. on Intelligent Robots and Systems*, 1995.

[3] D.P. Huttenlocher, A. Klanderman and J. Rucklidge, Comparing images using the hausdorff distance, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(9), 1993.

[4] D.P. Huttenlocher, W.J. Rucklidge and J.J. Noh, Tracking non-rigid objects in complex scenes, *Fourth Int. Conf. on Computer Vision*, 1993.

[5] S. Jiansho and C. Tomasi, Good features to track, *Conf. on Computer Vision and Pattern Recognition*, 1994.

[6] P. Lasserre, R. Murrieta Cid, and M. Briot. Le modèle nominatif de régions: segmentation couleur et identification de régions par analyse de couleur et de texture. *Sixteenth Gretsi Symposium on Signal and Images Processing, Grenoble*, 1997.

[7] R. Murrieta-Cid, P. Lasserre and M. Briot, Color segmentation in principal regions for natural outdoor scenes, *Third Workshop on Electronic Control and Measuring Systems, Toulouse*, 1997.

[8] I.K. Sethi and G.P.R. Sarvarayudu, Hierarchical classifier design using manual information, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1982.

[9] T.S.C Tan and J. Kittler, Colour texture analysis using colour histogram, *IEEE Proc.-Vision Image Signal Process.*, 141(6):403–412, december 1994.

[10] M. Unser, Sum and difference histograms for texture classification, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1986.